

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation :
Nom, prénom : MALKI KHALED		N° candidat :
Épreuve ponctuelle <input type="checkbox"/>	Contrôle en cours de formation <input checked="" type="checkbox"/>	Date : / /
Organisation support de la réalisation professionnelle GastonMedilab		
Intitulé de la réalisation professionnelle Situation 2 : Mise en place d'une haute disponibilité web		
Période de réalisation : Modalité : <input type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe		Lieu : Gaston Berger
Compétences travaillées <input type="checkbox"/> Concevoir une solution d'infrastructure réseau <input checked="" type="checkbox"/> Installer, tester et déployer une solution d'infrastructure réseau <input checked="" type="checkbox"/> Exploiter, dépanner et superviser une solution d'infrastructure réseau		
Conditions de réalisation¹ (ressources fournies, résultats attendus) Ressources fournies : 2 serveurs web (nginx), 1 serveur haproxy, 1 base de données sous Mariadb Résultat attendu : Site web hautement disponible		
Description des ressources documentaires, matérielles et logicielles utilisées² Ressources matérielles : 1 serveur administrable à distance par l'interface de VMware ESXi, 1 PC sous Windows 11 pro Ressources logicielles : 4 VM rocky linux, 2 serveurs nginx, 1 serveur mariadb, 1 serveur haproxy, VMware Workstation pro 17, VMware ESXi 6.5, MySQL		
Modalités d'accès aux productions³ et à leur documentation⁴ https://docs.google.com/document/d/1wl_6oNLLRgRm9AjGUyJBfciOuwYd0ZounYAJw7lqY/edit?usp=sharing https://docs.google.com/document/d/1A9TKUZiQcGpiQT4FF_ZTOUJJaj94zaGw/edit?usp=sharing&oid=100076809106133572429&rtpof=true&sd=true		

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Administration des systèmes et des réseaux » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

³ Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve.* ». Les éléments nécessaires peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

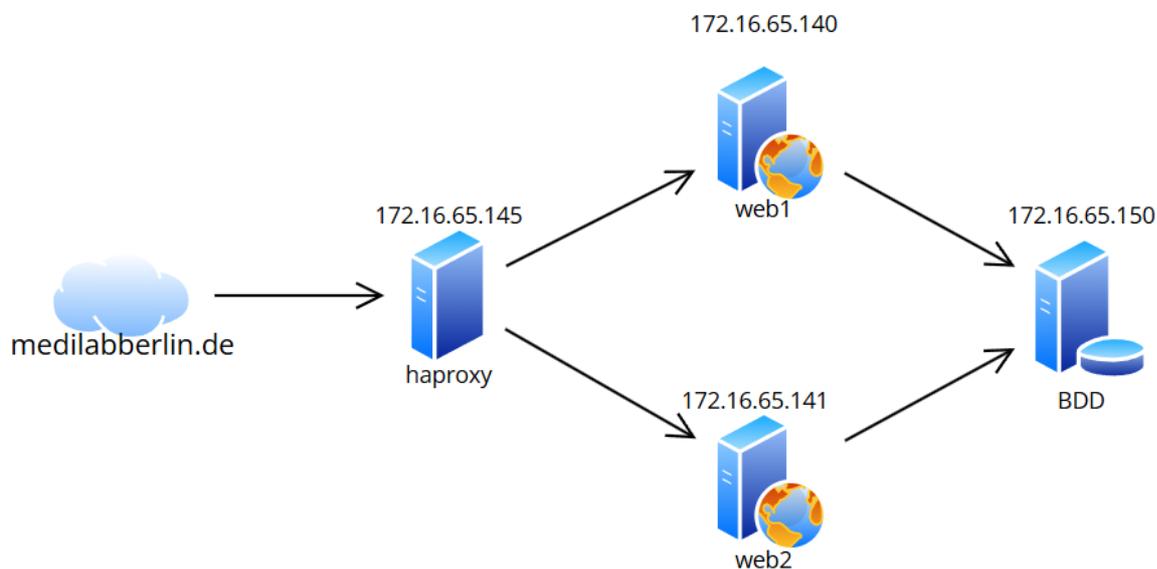
⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation, par exemples schéma complet de réseau mis en place et configurations des services.

Épreuve E6 - Administration des systèmes et des réseaux (option SISR)

**ANNEXE 7-1-A : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)**

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

Objectif : Avoir un site internet hautement disponible et certifié https. Et permettre la communication entre la base de données et les 2 serveurs web.



Prérequis : 4 VM rocky linux ; installation des services nginx, mariadb et haproxy ; installation php, MySQL et nano

Réalisation :

J'ai d'abord installé le service nginx et mariadb sur les 2 serveurs web, sans oublier les packages nécessaires à cette réalisation (php, mysql, nano). À la suite de cela, il a fallu configurer les firewalls de manière à autoriser les ports et protocoles utiles et un certificat https a été généré.

À la suite de l'installation de haproxy sur la vm rocky linux, on fera attention aux règles SELinux. La sécurisation du service haproxy se fait en partie en générant un certificat SSL. Enfin, les serveurs web y sont connectés.

La base de données fonctionnant grâce au service mariadb est connectée aux 2 serveurs web. Les connexions distantes depuis les serveurs web sont autorisées avec de hauts privilèges.

Protocole de test :

Afin de tester la haute disponibilité des serveurs web et le bon fonctionnement du haproxy, on éteint un des deux serveurs web. Puis on actualise la page web du site afin de vérifier l'accessibilité de celui-ci. On peut également vérifier le basculement sur le serveur haproxy. On fait cette manipulation 1 fois pour chaque serveur web.

Fichier de configuration de web1 :

```
GNU nano 5.6.1 site_commerce.fr.conf
server {
    listen 80;
    listen [::]:80;
    root /var/www/ecommerce.fr/PricingSubscription/;
    index index.html index.htm index.nginx-debian.html sign-up.php;
    server_name ecommerce.fr ;
    location ~* \.php$ {
        fastcgi_pass unix:/run/php-fpm/www.sock;
        include fastcgi_params;
        return 301 https://$host$request_uri/;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param SCRIPT_NAME $fastcgi_script_name;
    }
    access_log /var/log/nginx/access_ecommerce.fr.log;
    error_log /var/log/nginx/error_ecommerce.fr.log;
    location / {
        try_files $uri $uri/ =404;
    }
}

server {
    listen 443 http2 ssl;
    listen [::]:443 http2 ssl;
    ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
    ssl_dhparam /etc/ssl/certs/dhparam.pem;
    root /var/www/ecommerce.fr/PricingSubscription/;
    index index.html index.htm index.nginx-debian.html sign-up.php;
    server_name ecommerce.fr ;

    location ~* \.php$ {
        fastcgi_pass unix:/run/php-fpm/www.sock;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}
[ Read 41 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   ^U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line  ^-E Redo
```

```
server {
    listen 443 http2 ssl;
    listen [::]:443 http2 ssl;
    ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
    ssl_dhparam /etc/ssl/certs/dhparam.pem;
    root /var/www/ecommerce.fr/PricingSubscription/;
    index index.html index.htm index.nginx-debian.html sign-up.php;
    server_name ecommerce.fr ;

    location ~* \.php$ {
        fastcgi_pass unix:/run/php-fpm/www.sock;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param SCRIPT_NAME $fastcgi_script_name;
    }

    access_log /var/log/nginx/access_ecommerce.fr.log;
    error_log /var/log/nginx/error_ecommerce.fr.log;
    location / {
        try_files $uri $uri/ =404;
    }
}

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   ^U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line  ^-E Redo
```

BDD :

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| BERLINBDD |
| information_schema |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.033 sec)

MariaDB [(none)]>
```

Fichier de configuration de haproxy :

```
GNU nano 5.6.1 haproxy.cfg
-----
# Example configuration for a possible web application. See the
# full configuration options online.
#
# https://www.haproxy.org/download/1.8/doc/configuration.txt
#
-----
# Global settings
-----
global
# to have these messages end up in /var/log/haproxy.log you will
# need to:
#
# 1) configure syslog to accept network log events. This is done
#    by adding the '-r' option to the SYSLOGD_OPTIONS in
#    /etc/sysconfig/syslog
#
# 2) configure local2 events to go to the /var/log/haproxy.log
#    file. A line like the following can be added to
#    /etc/sysconfig/syslog
#
#    local2.*                               /var/log/haproxy.log
#
log      127.0.0.1 local2

chroot  /var/lib/haproxy
pidfile /var/run/haproxy.pid
maxconn 4880
user    haproxy
group   haproxy
daemon

# turn on stats unix socket
stats socket /var/lib/haproxy/stats

# utilize system-wide crypto-policies
ssl-default-bind-ciphers PROFILE-SYSTEM
ssl-default-server-ciphers PROFILE-SYSTEM
maxsslconn 256
tune.ssl.default-dh-param 2048
-----
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
-----
[ Read 105 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^I Execute    ^C Location   ^-U Undo      ^-A Set Mark  ^-] To Bracket ^-O Previous
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^G Go To Line  ^-R Redo      ^-G Copy      ^_ Where Was  ^-V Next
```

```
GNU nano 5.6.1 haproxy.cfg
tune.ssl.default-dh-param 2048
-----
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#-----
defaults
mode                http
log                 global
option              httplog
option              dontlognull
option http-server-close
option forwardfor   except 127.0.0.0/8
option              redispatch
retries              3
timeout http-request 10s
timeout queue       1m
timeout connect     10s
timeout client      1m
timeout server      1m
timeout http-keep-alive 10s
timeout check       10s
maxconn             3000

listen stats
bind *:::9999
mode http
stats enable
stats hide-version
stats uri /stats
stats admin if TRUE
stats auth haproxy:haproxy

#-----
# main frontend which proxys to the backends
#-----
frontend main
bind *:::80
acl url_static      path_beg    -i /static /images /javascript /stylesheets
acl url_static      path_end    -i .jpg .gif .png .css .js

use_backend static   if url_static
default_backend      app

frontend https_front
bind *:::443 ssl crt /etc/pki/tls/certs/haproxy.pem
default_backend web_servers

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^I Execute    ^C Location   ^-U Undo      ^-A Set Mark  ^-] To Bracket ^-O Previous
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^G Go To Line  ^-R Redo      ^-G Copy      ^_ Where Was  ^-V Next
```

```
#-----
# static backend for serving up images, stylesheets and such
#-----
backend static
balance leastconn
server static 127.0.0.1:4331 check

#-----
# round robin balancing between the various backends
#-----
backend app
balance roundrobin

backend web_servers

server web1 172.16.65.140:443 ssl verify none check
server web2 172.16.65.141:443 ssl verify none check

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^I Execute    ^C Location   ^-U Undo      ^-A Set Mark  ^-] To Bracket ^-O Previous
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^G Go To Line  ^-R Redo      ^-G Copy      ^_ Where Was  ^-V Next
```

